WHAT IS CLAIMED IS:

1. A method of managing device driver tokens, comprising:

receiving a request to communicate with a device associated with an x bit device card, the x bit device card being associated with a device driver, and the request being received on an x+ bit computer;

creating an x+ bit pointer corresponding to the request, the x+ bit pointer having a value based on the x+ bits;

storing data corresponding to the request in a memory associated with the driver at the address corresponding to the x+ bit pointer;

passing the request to the device card by creating a first control block having an x bit token field, the creating comprising converting the x+ bit pointer to an x bit token having a y bit token associated with the x+ bit pointer, and a z bit word corresponding to an error checking identifier;

receiving a response from the device card upon completion of the request, the response having a second control block, the second control block having the x bit token;

determining if error checking is requested;

if error checking is requested, then validating the x bit token; and

if the x bit token is valid, then:

converting the y bit token back into the x+ bit pointer; and

passing the x+ bit pointer back to the driver to release the address corresponding to the x+ bit pointer.

2. The method of claim 1, wherein x comprises 16 bits, and x+ comprises 32 bits.

3. The method of claim 1, wherein x comprises 32 bits, and x+ comprises 64 bits.

4. The method of claim 1, wherein said validating the x bit token comprises using the z bit word for parity checking.

5. The method of claim 1, wherein said validating the x bit token comprises using the z bit word in a check sum algorithm.

6. The method of claim 1, wherein said converting the x+ bit pointer to a y bit token and a z bit word comprises:

using a hash function to convert the x+ bit pointer into a unique y bit token; and

storing the unique y bit token in a hash bucket as determined by the hash function.

7. The method of claim 1, further comprising if the x+ bit pointer is not valid, then preventing the address associated with the x+ bit pointer from being accessed.

8. A method of managing device driver tokens, comprising:

receiving a request to communicate with a device being associated with an x bit device card and a device driver, the request being received on an x+ bit computer, and x+ being greater than x;

creating an x+ bit pointer corresponding to the request;

storing data corresponding to the request in a memory at the address corresponding to the x+ bit pointer;

passing the x+ bit pointer into a hash function to generate a y bit token;

generating a z bit word in accordance with a requested level of error checking;

creating an x bit request control block comprising the y bit token and the z bit word, where x=y+z;

passing the x bit request control block to the device card;

in response to receiving a response from the device card upon completion of the request, creating a response control block having the x bit token;

if z<>0, then using z to validate the x bit token;

if x is valid, then:

using the hash function to convert the y bit token into an x+ bit pointer;

passing the x+ bit pointer back to the driver; and

releasing the address corresponding to the x+ bit pointer.

9. The method of claim 8, wherein said validating the x bit token comprises using the z bit word for parity checking.

10. The method of claim 8, wherein said validating the x bit token comprises using the z bit word in a check sum algorithm.

11. The method of claim 8, further comprising if the x+ bit pointer is not valid, then preventing the address associated with the x+ bit pointer from being accessed.

12. The method of claim 8, wherein the memory is associated with the driver.

13. A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, cause the processor to perform the following:

receive a request to communicate with a device associated with an x bit device card, the x bit device card being associated with a device

driver, and the request being received on an x+ bit computer;

create an x+ bit pointer corresponding to the request, the x+ bit pointer having a value based on the x+ bits;

store data corresponding to the request in a memory associated with the driver at the address corresponding to the x+ bit pointer;

pass the request to the device card by creating a first control block having an x bit token field, the creating comprising converting the x+ bit pointer to an x bit token having a y bit token associated with the x+ bit pointer, and a z bit word corresponding to an error checking identifier;

receive a response from the device card upon completion of the request, the response having a second control block, the second control block having the x bit token;

determine if error checking is requested;

if error checking is requested, then validate the x bit token; and

if the x bit token is valid, then:

convert the y bit token back into the x+ bit pointer; and

pass the x+ bit pointer back to the driver to release the address corresponding to the x+ bit pointer.

14. The machine-readable medium of claim 13, wherein the processor validates the x bit token by using the z bit word for parity checking.

15. The machine-readable medium of claim 13, wherein the processor validates the x bit token by using the z bit word in a check sum algorithm.

16. The machine-readable medium of claim 13, wherein the processor converts the x+ bit pointer to a y bit token and a z bit word by:

using a hash function to convert the x+ bit pointer into a unique y bit token; and

storing the unique y bit token in a hash bucket as determined by the hash function.

17. The machine-readable medium of claim 13, further comprising if the x+ bit pointer is not valid, then the processor prevents the address associated with the x+ bit pointer from being accessed.

18. An apparatus for managing a device, comprising:

a device driver associated with an x+ bit computer to handle a request to communicate with a device associated with an x bit device card by:

creating an x+ bit pointer; and

storing information about the request at an address corresponding to the x+ bit pointer in a memory associated with the device driver;

a translator to convert the x+ bit pointer to an x bit token, the converting including:

generating a y bit token; and

generating a z bit word representing a level of error checking to validate the y bit token;

the device driver to further:

create a request control block comprising the x bit token;

to send the request control block to the x bit device card for processing;

to receive a response control block comprising the x bit token from

the device card; and

to validate the x bit token of the response control block in accordance with the level of error checking determined by the z bit word; and

the translator to further convert the y bit token into the x+ bit pointer if the x bit token is valid; and

the device driver to further release the memory at the address associated with the x+ bit pointer if the x bit token is valid.

19. The apparatus of claim 18, wherein the translator is integrated with the device driver.

20. A system for robust device driver token management, comprising:

a device driver associated with an x+ bit computer to process a request to communicate with a device associated with an x bit device card by creating an x+ bit pointer;

a device driver memory to store data associated with the request at the address corresponding to the x+ bit pointer;

a robust translator to convert the x+ bit pointer to an x bit token comprising a y bit token associated with the x+ bit pointer, and a z bit word for validating the y bit token;

the device driver to further create a request control block comprising the x bit token, and to send the request control block to the x bit device card;

the x bit device card to process the request control block, to create a response control block in response to processing the request control block, and to send the response control block to the device driver;

the device driver to further validate the x bit token of the response control

block in accordance with the level of error checking determined by the z bit identifier;

the translator to further convert the y bit token into the x+ bit pointer if the x bit token is valid; and

the device driver to further release the memory at the address associated with the x+ bit pointer if the x bit token is valid.

21. The system of claim 20, further comprising:

a request queue to hold the request control block for processing by the device card; and

a response queue to hold the response control block for processing by the device driver.

22. The system of claim 20, wherein the device driver validates the x bit token by using the z bit word for parity checking.

23. The system of claim 20, wherein the device driver validates the x bit token by using the z bit word in a check sum algorithm.